

Adaptive Brandsimulation mit FEM

Tag der Forschung 2015

04.02.2015 | Kevin Drzycimski | Jülich Supercomputing Centre



Motivation

Brandsimulation

JuFire

Adaptive Gitter
Zeitverfahren

Demonstration

Ausblick



Motivation

Brandsimulation

JuFire

Adaptive Gitter
Zeitverfahren

Demonstration

Ausblick

Motivation

Problem



1

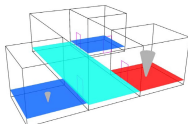
Brandschutz

- Leistungsorientierte Nachweise zur Personensicherheit im Brandfall
- Platzierung und Dimensionierung anlagentechnischer Brandschutzkomponenten
- Standsicherheitsnachweis / Heißbemessung

¹Wikimedia Commons

Motivation

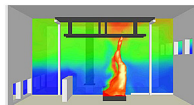
Lösung



2



3



4

- Theorie
- Experiment
- Simulation

²fire-simulation.at

³I.F.I. Institut für Industrieaerodynamik GmbH

⁴nist.gov



Motivation

Brandsimulation

JuFire

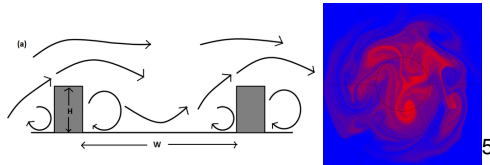
Adaptive Gitter
Zeitverfahren

Demonstration

Ausblick

Brandsimulation

Einleitung

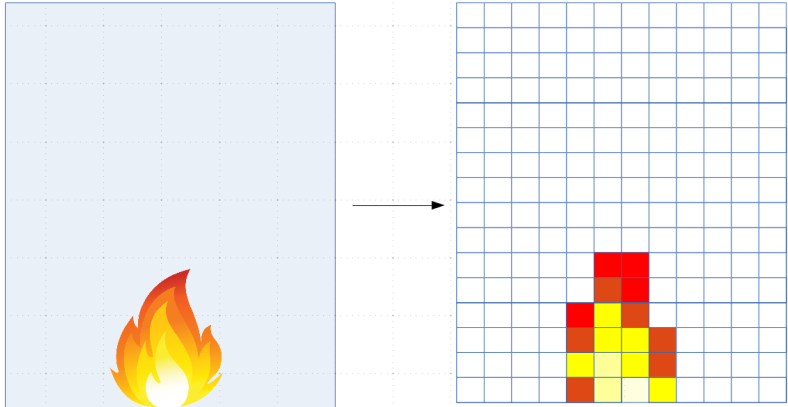


- Strömung von Rauch und Heißgasen ist Teil der Fluidodynamik
 - Grundlage: *Navier-Stokes-Gleichungen*
 - Geschlossene Lösung nur für simple Fälle
- ⇒ Computational Fluid Dynamic (CFD)

⁵Wikimedia Commons

Brandsimulation

Diskretisierung





Brandsimulation

FDS

Fire Dynamics Simulator

Pro

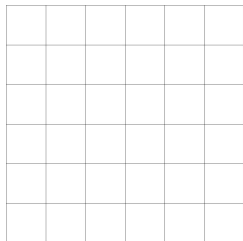
- Freie Software des NIST
- Ein in der Community anerkanntes Werkzeug, Quasi-Standard
- Großer Funktionsumfang

Contra

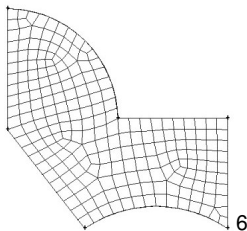
- Statische, äquidistante Gitter
- Nicht geeignet fuer Parallelisierung

Brandsimulation

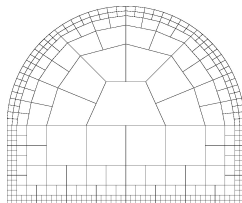
Gitter



äquidistant



unstrukturiert



hängende Knoten

⁶ANSYS



Motivation

Brandsimulation

JuFire

Adaptive Gitter
Zeitverfahren

Demonstration

Ausblick



JuFire

Ziele

Thesis

- Simulation von Rauch
- Adaptivität ohne Nutzerinteraktion
- Lauffähig auf Workstations, Clustern und Supercomputern

Langfristig

- Strahlung
- Verbrennung



JuFire

Übersicht

- Während der Promotion entstehendes Software-Paket
- Open-Source, C++
- Baut auf weiteren Paketen auf:
 - deal.II** Finite-Elemente-Bibliothek
 - p4est** Lastverteilung auf Großrechnern
 - Trilinos** Lineare Algebra, Gleichungssystemlöser

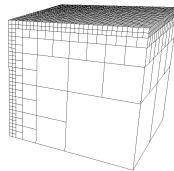
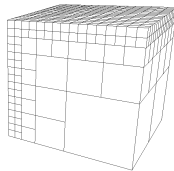
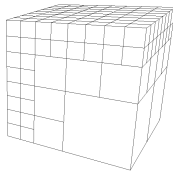
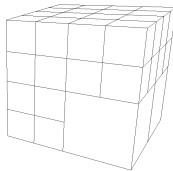
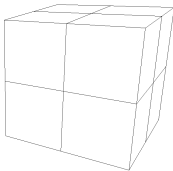


Adaptive Gitter

- Hohe Gitterauflösung = hohe Genauigkeit, aber auch mehr Speicher- und Rechenaufwand
- Verfeinere Zellen nicht global, sondern nur an interessanten Stellen

Adaptive Gitter

- Hohe Gitterauflösung = hohe Genauigkeit, aber auch mehr Speicher- und Rechenaufwand
- Verfeinere Zellen nicht global, sondern nur an interessanten Stellen



global
adaptiv

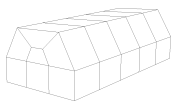
64
50

512
246

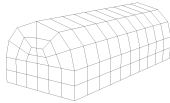
4096
1086

32768
4558

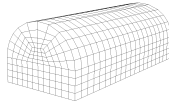
Adaptive Gitter



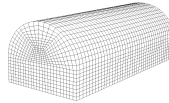
global
adaptiv



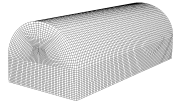
280
84



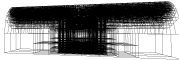
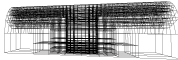
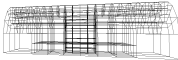
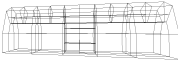
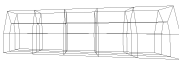
2240
504

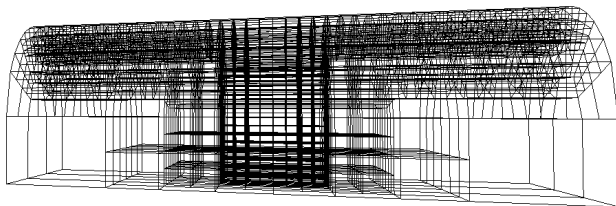
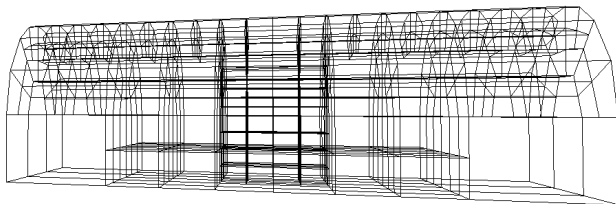
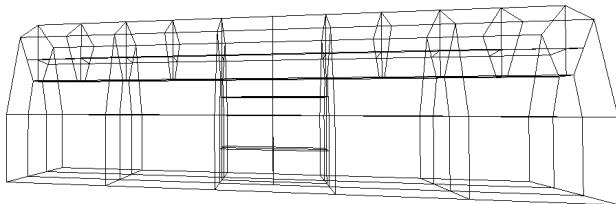


17920
3430



143360
23982







Adaptive Gitter

- Bisher: Gitter mit Vorwissen erstellt
- Nicht sinnvoll bei komplexen Geometrien oder stark ändernden Lösungen
- ⇒ Adaption der Gitter zur Laufzeit anhand Indikatoren und Fehlerschätzer
- z.B. $\nabla \mathbf{u}$, ∇T , $\dot{\mathbf{u}}$, lokale Projektion mit höherer Fehlerordnung, ...
- ⇒ automatische Verfeinerung und Vergröberung = bessere Ausnutzung der Rechnerkapazität

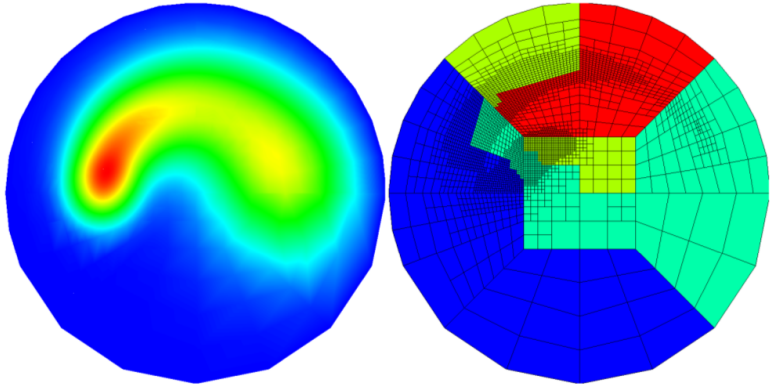
Adaptive Gitter

Algorithmus

- 1 Berechne für jede Zelle die Indikatoren
- 2 Normalisieren auf $[0, 1]$
- 3 Markiere die oberen $t_{refine}\%$ zum Verfeinern
Markiere die unteren $t_{coarse}\%$ zum Vergröbern
- 4 Ausführen des Verfeinerns/Vergröbern
- 5 Bei verteiltem Speicher Gitter ausbalancieren
- 6 Lösungen auf neues Gitter interpolieren

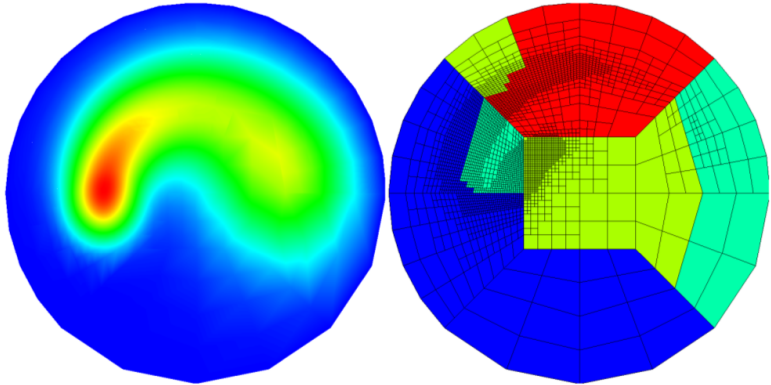
Adaptive Gitter

Parallel AMR 1



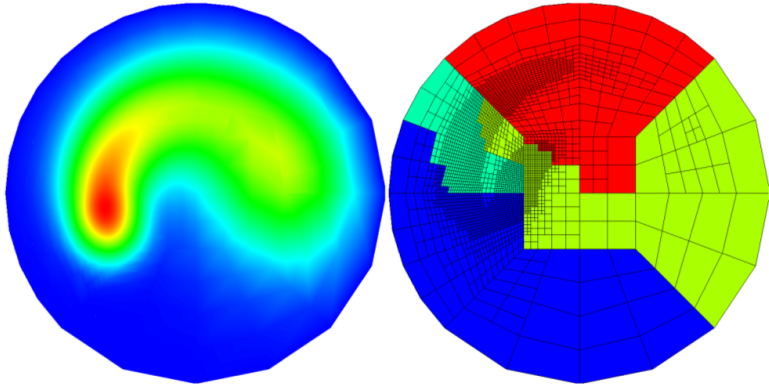
Adaptive Gitter

Parallel AMR 2



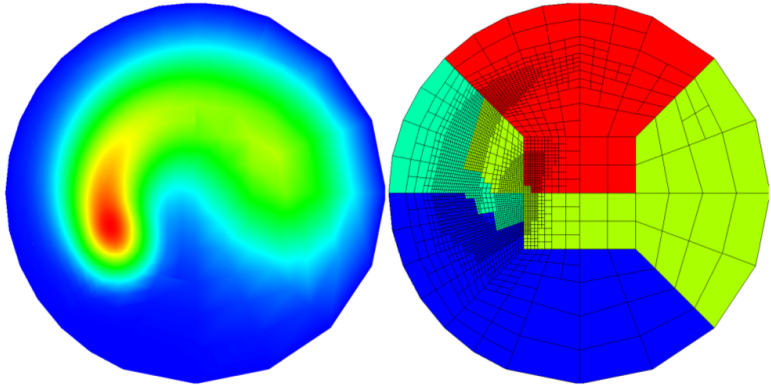
Adaptive Gitter

Parallel AMR 3



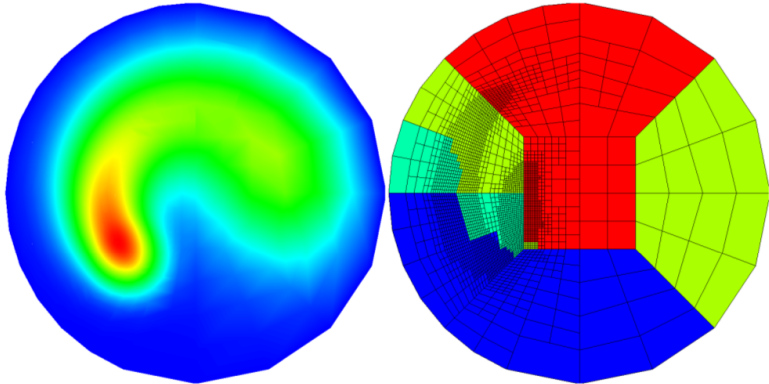
Adaptive Gitter

Parallel AMR 4



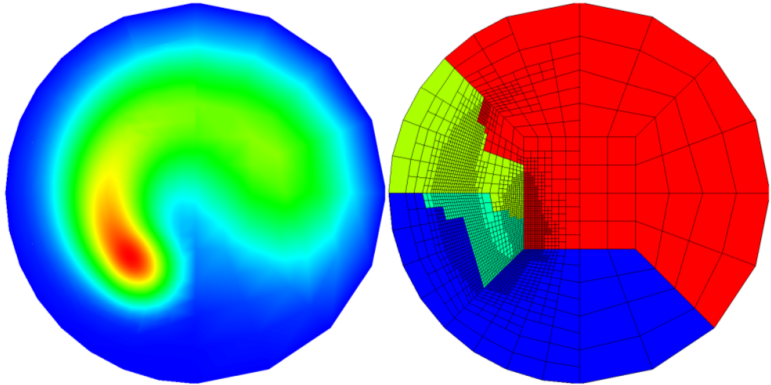
Adaptive Gitter

Parallel AMR 5



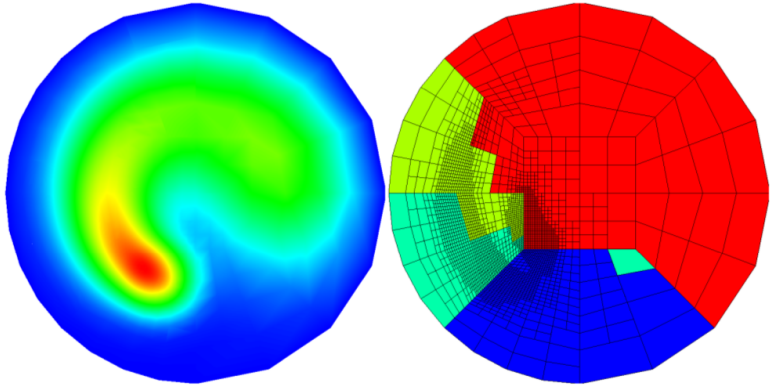
Adaptive Gitter

Parallel AMR 6



Adaptive Gitter

Parallel AMR 7



Zeitschrittverfahren

Inkompressibel

- Geschwindigkeiten niedriger als Schall
- Variation der Dichte sehr gering

⇒ *Boussinesq-Approximation*

$$\dot{\rho} = 0$$

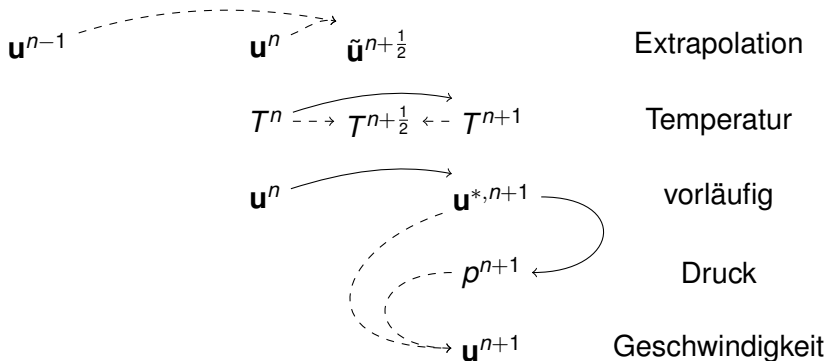
- Inkompressible Navier-Stokes-Gleichungen:

$$\dot{\mathbf{u}} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = f(T)$$

$$\nabla \cdot \mathbf{u} = 0$$

$$\dot{T} + \mathbf{u} \cdot \nabla T - \nu \Delta T = q$$

Zeitschrittverfahren Splitting





Motivation

Brandsimulation

JuFire

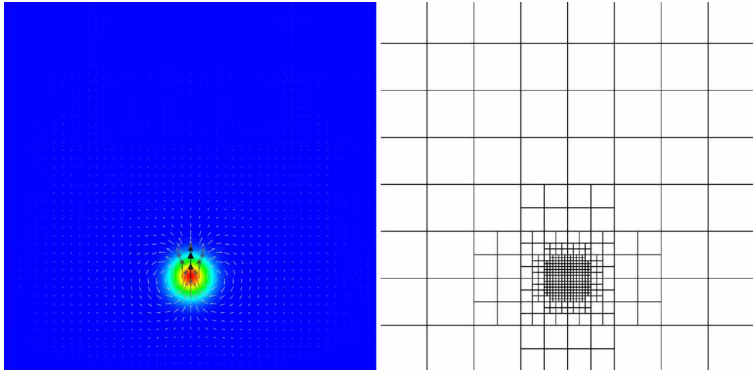
Adaptive Gitter
Zeitverfahren

Demonstration

Ausblick



Demonstration





Motivation

Brandsimulation

JuFire

Adaptive Gitter
Zeitverfahren

Demonstration

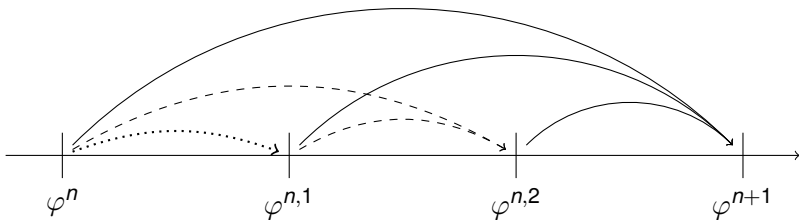
Ausblick

Ausblick

Numerik

Runge-Kutta-Verfahren höherer Ordnung

- Crank-Nicolson ist Einschritt-Verfahren 2. Ordnung
- Mehrere Zwischenstufen \Rightarrow höhere Fehlerordnung
- Schrittweitensteuerung durch eingebettete Verfahren





Ausblick

Anwendung

Validierung

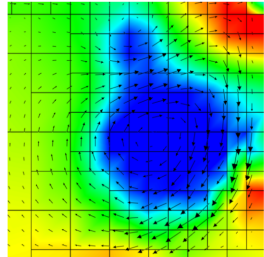
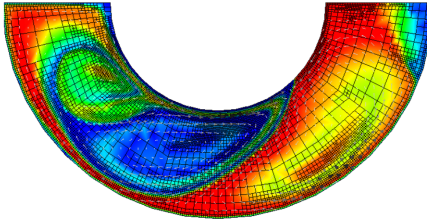
- Korrekt?

⇒ Vergleich mit realen Experimenten

Ende

Vielen Dank für Ihre Aufmerksamkeit

- Haben Sie noch Fragen?





BERGISCHE
UNIVERSITÄT
WUPPERTAL



Backup

Zeitschrittverfahren

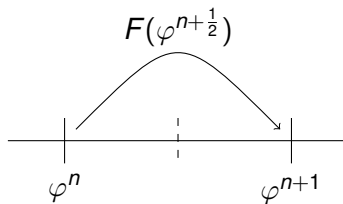
Crank-Nicolson

- Allgemeine Differentialgleichung für φ , Zeitänderung hängt von Kraft F ab

$$\dot{\varphi} = F(\varphi)$$

- Von Zeit n nach $n+1$ verwende Kraft zum Halbschritt $n+\frac{1}{2}$

$$\frac{\varphi^{n+1} - \varphi^n}{h} = F(\varphi^{n+\frac{1}{2}})$$

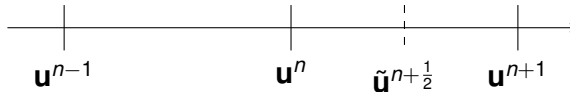


Zeitschrittverfahren

Time-Splitting und Extrapolation

- Von (\mathbf{u}^n, p^n, T^n) nach $(\mathbf{u}^{n+1}, p^{n+1}, T^{n+1})$ zu aufwändig
- \Rightarrow Time-Splitting, Größen separieren
- Aus \mathbf{u}^n und \mathbf{u}^{n-1} Näherung für $\mathbf{u}^{n+\frac{1}{2}}$ extrapolieren:

$$\tilde{\mathbf{u}}^{n+\frac{1}{2}} = \frac{3}{2}\mathbf{u}^n - \frac{1}{2}\mathbf{u}^{n-1}$$



Zeitschrittverfahren

Temperatur

- Crank-Nicolson Schritt mit Temperatur:

$$\dot{T} + \mathbf{u} \cdot \nabla T - \nu \Delta T = q$$

$$\frac{T^{n+1} - T^n}{h} + \tilde{\mathbf{u}}^{n+\frac{1}{2}} \cdot \nabla \frac{T^{n+1} + T^n}{2} - \nu \Delta \frac{T^{n+1} + T^n}{2} = q^{n+\frac{1}{2}}$$

- Mit neuem T^{n+1} Näherung $T^{n+\frac{1}{2}}$ interpolieren:

$$T^{n+\frac{1}{2}} = \frac{1}{2} T^{n+1} + \frac{1}{2} T^n$$

Zeitschrittverfahren

Geschwindigkeit, Druck

- Mit diesem $T^{n+\frac{1}{2}}$ wird eine vorläufige Geschwindigkeit $\mathbf{u}^{*,n+1}$ berechnet:

$$\frac{\mathbf{u}^{*,n+1} - \mathbf{u}^n}{h} + (\tilde{\mathbf{u}}^{n+\frac{1}{2}} \cdot \nabla) \frac{\mathbf{u}^{*,n+1} + \mathbf{u}^n}{2} - \nu \Delta \frac{\mathbf{u}^{*,n+1} + \mathbf{u}^n}{2} = f(T^{n+\frac{1}{2}})$$

- Druck p^{n+1} über Laplace-Gleichung:

$$-\Delta p^{n+1} = -\frac{1}{h} \nabla \cdot \mathbf{u}^{*,n+1}$$

- Schließlich richtige Geschwindigkeit per Korrektur:

$$\mathbf{u}^{n+1} = \mathbf{u}^{*,n+1} - h \nabla p^{n+1}$$

Ausblick

Numerik

Incremental Pressure Correction Scheme

- Idee: Druckgradienten bei \mathbf{u}^* berücksichtigen
- Danach lediglich Druck-Korrektur mit der Laplace-Gleichung
- Fehlerterm: $O(h^2)$

$$\begin{aligned}\frac{\mathbf{u}^{*,n+1} - \mathbf{u}^n}{h} + \dots &= f(T^{n+\frac{1}{2}}) - \nabla p^n \\ -\Delta \psi^{n+1} &= -\frac{1}{h} \nabla \cdot \mathbf{u}^{*,n+1} \\ \mathbf{u}^{n+1} &= \mathbf{u}^{*,n+1} - h \nabla \psi^{n+1} \\ p^{n+1} &= p^n + \psi^{n+1}\end{aligned}$$

Backup

- Normales FEM nicht geeignet für Konvektion
- ⇒ Stabilisierung
- Streamline-Diffusion mit Taylor-Galerkin

$$\dot{u} + \beta \cdot \nabla u - \nu \Delta u = f$$

$$(u, v) + h(\beta \cdot \nabla u, v) + h(\nu \nabla u, \nabla v) + \frac{h^2}{6}(\beta \cdot \nabla u, \beta \cdot \nabla v) = (u^{\text{old}}, v) + \dots$$

Backup

- Nichtlinearer Advektionsterm $\mathbf{u} \cdot \nabla \mathbf{u}$
- Schief-symmetrische Form

$$c(\mathbf{w}, u, v) = (\mathbf{w} \cdot \nabla u, v) + \frac{1}{2}(\nabla \cdot \mathbf{w} u, v)$$

- Verwende $c(\mathbf{u}^{*,n+\frac{1}{2}}, \mathbf{u}^{n+1}, \mathbf{v})$